

V. Claims

We claim:

1. A method for continuing a preorder traversal of a binary tree formed by a family of nodes, each node in the family comprising a child pointer, a sibling pointer, and a unique counter value, the method comprising:

receiving as input a continuation node and a lineage for the continuation node, the continuation node and lineage comprising an output of a first partial preorder traversal of the binary tree;

locating an updated continuation node in the binary tree by comparing the unique counter values of the family of nodes to a current node, the continuation node comprising the first current node; and

continuing the preorder traversal of the binary tree from the updated continuation node.

2. The method of claim 1 wherein the comparing step follows the lineage of the continuation node.

3. The method of claim 1 wherein the binary tree comprises a dynamic binary tree.

4. The method of claim 1 wherein the binary tree represents a general tree.

5. The method of claim 1 wherein the binary tree represents a family of related processes.

6. The method of claim 1 wherein the binary tree represents a disk file directory structure.

7. The method of claim 1 wherein the binary tree represents a computer program structure.
8. The method of claim 1 wherein the nodes in the family further comprise a parent pointer.
9. The method of claim 1 wherein an abbreviated continuation node lineage is received as the input.
10. The method of claim 1 wherein the binary tree resides in a first environment and the input is received from a second distinct environment.
11. A computer-readable medium having stored thereon a data structure for managing a plurality of elements related by hierarchy, the data structure representing one of the elements and comprising:
 - a data value field, a child pointer field, a sibling pointer field, and a unique counter field.
12. The computer-readable medium of claim 11 wherein the data structure further comprises a parent pointer field.
13. The computer-readable medium of claim 11 wherein the elements comprise nodes in a tree.
14. The computer-readable medium of claim 11 wherein the elements comprise nodes in a binary tree.
15. The computer-readable medium of claim 11 wherein the unique counter field is populated with a non-decreasing counter value.

16. The computer-readable medium of claim 11 wherein the elements represent a process family structure.
17. The computer-readable medium of claim 11 wherein the elements represent a disk file directory structure.
18. The computer-readable medium of claim 11 wherein the elements represent a computer program structure.
19. A method for locating an updated continuation node in a dynamic binary tree formed by a family of nodes, each node in the family comprising a child pointer, a sibling pointer, and a unique counter value, the method comprising:
- (a) receiving as input a continuation node and an abbreviated lineage for the continuation node; and
 - (b) traversing the nodes in the tree along the abbreviated lineage until the counter value indicates that a first valid node beyond the continuation node has been reached.
20. The method of claim 19 further comprising:
- (b)(1) determining whether the continuation node still exists in the tree; and
 - (b)(2) if the result of step (b)(1) is no, traversing the nodes in the tree along the abbreviated lineage until the counter value indicates that a first valid node beyond the continuation node has been reached.
21. The method of claim 19 wherein step (b) comprises:
- (b)(1) determining whether the continuation node still exists in the tree;
 - (b)(2) if the result of step (b)(1) is no, determining whether the continuation node has a depth equal to or less than zero; and
 - (b)(3) if the result of step (b)(2) is no, traversing the nodes in the tree along the abbreviated lineage until the counter value indicates that a first valid node beyond the

continuation node has been reached.

22. The method of claim 19 wherein steps (a) and (b) are performed in a first data environment and the input is received from a second process in a second non-native data environment.
23. The method of claim 22 further comprising:
 - (c) passing as an output the updated continuation node to the second process.
24. The method of claim 19 wherein step (a) comprises:
 - (a)(1) receiving as input a continuation node and a lineage for the continuation node; and
 - (a)(2) extracting an abbreviated lineage for the continuation node from the continuation node lineage.
25. The method of claim 19 wherein the binary tree represents a general tree.
26. The method of claim 19 wherein the binary tree represents a family of related processes.
27. The method of claim 19 wherein the binary tree represents a disk file directory structure.
28. The method of claim 19 wherein the binary tree represents a computer program structure.
29. The method of claim 19 wherein the nodes in the family further comprise a parent pointer.

